Requirements Management for the Oceanographic Information System at the Naval Oceanographic Office

John A. Lever
Naval Oceanographic Office
Stennis Space Center, MS
leverj@navo.navy.mil
Michael Elkins
Northrop Grumman Information Technology
Stennis Space Center, MS
michael.elkins@ngc.com

Abstract - ***Recognizing the complexity of the interdependencies in a diverse information system, NAVOCEANO is conducting a rigorous overhaul of the Requirements Management process to standardize, develop, and maintain the requirements that the OIS is intended to satisfy. The intended result is an integrated Requirements Enterprise Architecture Documentation (READ) suite for the OIS, expressed in database and document form that promotes interoperability and fiscal responsibility of IT acquisition. Once the requirements baseline is established, the requirements management process will provide a mechanism to establish, delete, and change requirements. To accomplish this result, our Systems Engineers have decomposed organizational policies, studies, and directives to extract general requirements and are soliciting detailed requirements from system users. Each requirement is linked to our existing Enterprise Architecture. The current Configuration Management (CM) process will be modified to account for the new Requirements Management process to ensure the validity of the READ suite. This paper will describe the process used, high level requirements documented, and the lessons learned in Requirements Management.***

The Naval Oceanographic Office (NAVOCEANO) is a third-echelon Naval command whose primary purpose is the collection and processing of data relating to physical properties of the Earth's oceans. Inherent in this mission is a requirement to ingest, store, process, and archive massive quantities of data. The Office has mission applications in numerous disciplines including hydrography, bathymetry, physical oceanography, acoustics, geophysics, and others. The Oceanographic Information System (OIS) is the primary information technology domain for scientific processing at NAVOCEANO; it consists of more than 3000 thousand scientific processing systems that incorporate over 3 million lines of custom computer code working with numerous commercial off the shelf (COTS) applications.

I. Introduction Center

The Naval Oceanographic Office (NAVOCEANO) has the responsibility to collect and process data relating to the physical properties of the Earth's oceans. Mission applications include hydrography, bathymetry, physical oceanography, acoustics, geophysics, bioluminescence, environmental, and biological. As the primary information technology domain for scientific processing at NAVOCEANO, the Oceanographic Information System (OIS) is responsible for ingesting, analyzing, processing, storing, and archiving data from over 3000 scientific processing systems which utilize over 3 million lines of custom computer code and hundreds of commercial off the shelf (COTS) applications.

In support of NAVOCEANO's varying mission requirements over the course of time, numerous independent information systems evolved. The logical step for the NAVOCEANO computing enterprise was a movement to consolidate and eliminate legacy systems. In 1994 the Commander, Naval Meteorology and Oceanography Command approved three unique information systems that incorporated all functions from the numerous discipline-oriented systems. The result included the OIS, a roll-up of scientific processing systems with the addition of a few "special-interest" subsystems.

In order to manage the organization's numerous diverse systems effectively, the CIO established an Enterprise Architecture (EA) that catalogs the numerous OIS sub-systems, their components, and their interdependencies with other systems and subsystems. (Hereafter, the term "systems" will be used to denote the distinct subsystems of the OIS.) Establishing the EA was the first part of the OIS roadmap to implementing sound Information Management Engineering. The necessary and subsequent part of this process was to gather document, and relate functional requirements to these systems. Inherent in this activity was to create a process that keeps the baseline current with changes in requirements and systems. This paper describes

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **01 SEP 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Requirements Management for the Oceanographic Information System at the Naval Oceanographic Office**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Oceanographic Office Stennis Space Center, MS**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

See also ADM002146. Oceans 2003 MTS/IEEE Conference. Held in San Diego, California on September 22-26, 2003. U.S. Government or Federal Purpose Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**4**	

the methods used for requirements capture and maintenance as an integral function of information engineering.

II. Capturing Necessary Data Center

Our (EA) was documented in a Microsoft Access Database, and in order to effect a smooth integration we determined that using Access to capture and manage our requirements was the most efficient use of resources. Once integrated, we had plans to migrate our entire EA inclusive of requirements to a more robust database with a Web interface. The first task for our requirements analysts was to determine the type, level, and details of the requirements that needed to be captured in order for us to effectively associate these with our systems. We used the same approach teams when developing a statistical survey. First, determine the type of data you need and then derive the questions to support that statistical analysis. In our case, we looked at the kind of data that would support an effective engineering process and derived the supporting requirement fields. Some of the queries to make :

- If a requirement is changed or one is added:
    - What systems are affected?
    - What alternative systems can support the requirement?
- If a system is down for maintenance, is damaged, or destroyed:
    - What is the impact on the requirements?
    - Which organizations are affected?
- If an application/platform/operating system is being considered for upgrade:
    - What requirements may be affected?
    - What organizations may be affected?

In addition to these queries, it is often useful to be able to sort data by a variety of criteria. Based upon the kind of information that we wanted to extract and the way we wanted to sort this data, we built an application to capture the following data for each requirement.

- Status
    - Current
    - Obsolete
    - Deferred
    - Proposed
    - Unvalidated
- Category
    - Qualitative
    - Quantitative
- Priority
    - Mandatory
    - Preferred
- Requirement Date
- Requirement Type
    - Functional
    - Performance
    - Environmental
    - Interface
    - Quality
    - Constraint
    - Security
- Champion
- Key Areas
- Requirement Description
- Requirement Comments
- Related Requirements
- Requirement Sources
- Related Systems
- Point of Contact

III. Application Design Center

Once the data structure was defined, our team set out to build an application that would support not only the initial documentation of the requirements but also continued maintenance and configuration control of these requirements once captured. A method for keeping requirement currency was essential to this effort. A snapshot of requirements as they existed during the course of this effort would be outdated within a month if there were not a method for keeping the database current. The goal was to create a user interface that was intuitive and facilitated change. A challenge to our effort was the knowledge that requirements are often nested in varying degrees of granularity as a qualitative requirement is broken down into its many quantitative parts. The application had to be able to support nested requirements and maintain and display their relationship. Initially, we tried to determine the specific number of levels that we would need, but as soon as we set an artificial boundary, we would find a case where we needed to extend below this limit. We altered the tool to accommodate for an unlimited number of levels.

IV. Requirement Bundling Center

A single requirement cannot be fully understood when viewed in isolation. To fully understand the context of the requirement, the user must be able to determine related requirements and the context of the requirement. To achieve this, our team adopted one of the proven Information Engineering methodologies and used it as a baseline for organizing the system requirements. Adhering to the practices of the Integration Definition for Function (IDEF0) modeling as described in Federal Information Processing Standard 183 (December 21, 1993), our team determined our top level requirements. A basic IDEF0 chart is illustrated below.

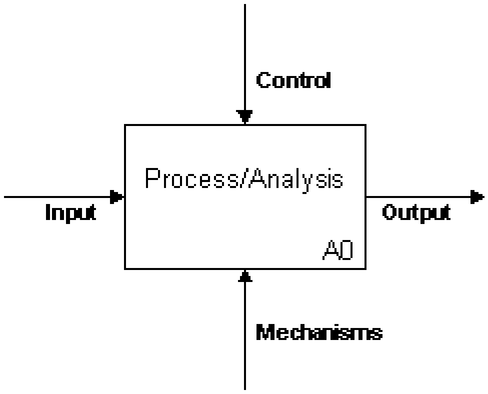

Fig. 1. IDEF0 Model

Our requirements effort represented the IDEF0 elements of Input, Process, Analysis, Output, and Mechanisms with the following level one requirements:

- The system shall collect data from sources via methods (input)
- The system shall process data to produce products (process)
- The system shall analyze data for a specified purpose (analysis)
- The system shall disseminate products to destinations via methods (output)
- The system shall store and manage data and products (mechanisms)

Through the segregation of requirements into these specific activities and the ability to associate requirements with other requirements, our team was able to effectively capture and relate requirements in a meaningful manner.

V. Top Down Data Collection Center

To orient our team with the functional requirements and to minimize the time required from our requirement champions, we initially used existing organizational documentation as the source of requirements. We extracted requirements from documents including Mission Needs statements, Operational Requirements documentation, Organizational Manuals, Program Management Plans, Configuration Management Board Charters, Project Studies, Project Fact Sheets, and Systems Test Plans. This analysis gave us a baseline of general requirements that allowed us to refine our requirement hierarchy. One drawback to this approach was that the accuracy of some requirements was possibly diminished due to the age of some of the documents. A flexible database and the capability to easily move and delete requirements as they were validated proved essential.

VI. Bottom Up Data Collection Center

With general requirements and organizational structure defined, our analysts branched out to begin a bottom up review of requirements. The first step in this process was to brief the project, its purpose, and its goals to representatives from each of the eight organizational configuration management boards. The next step in this process was to meet with each board chairman individually and discuss the projects that fall under his/her organization and obtain points of contact (POC) for each product area. Our final step was to set up interviews with each POC to discuss the elements of his/her process. Preceding each interview, the POC was sent a list of the projects that were within his/her domain area, a description of our approach, and any existing requirements that we had documented from our top down review. Interviews varied in length from 1 to 3 hours each. Once an interview was completed, we would immediately document the findings within the READ system. Once documented, we would return to each POC with the specified requirements and review them for validation.

VII Change Control Center

Once the interviews are completed and all the requirements are validated, we will consider the system baselined and will implement our Requirements Change Process. When the configuration management board associated with a particular requirement makes a change to a requirement or to a system that relates to a requirement, they will use the READ system to determine the impact on the related systems or requirements. Any necessary changes/additions/deletions of requirements can be directed by the board. When a change of any kind is made to a requirement, the requirement version is incremented. The user making the change specifies if this is a major revision or a minor revision and comments on the nature of the change. The change details are stored in the database, and a change history report is available to READ users.

VIII. Lessons Learned Center

Although this is an on-going effort, we have gained some valuable insights as we have gone through this process. These lessons include the difficulty of extracting requirements from diverse documentation of varying vintage, the value of corporate commitment (organizational and champions), and the importance of an organized approach to grouping and relating requirements.

A. Extracting Requirements From Existing Documentation

As specified earlier, we had a diverse set of documents that we used to establish our general requirements. It was a considerable challenge to analyze these documents and try to extract requirements in a standardized method. Each of these supporting documents stated requirements in their own unique way and many do not easily translate into our standard. We also found that very thorough studies and analysis that occurred as little as two years earlier were often outdated due to technological or organizational changes. It became apparent through this effort that not only is it important to establish a format for documenting requirements but also most efforts are only a snapshot of requirements for a given organization at a given time. This stressed the importance of implementing a requirements management process upon completion of the requirements baseline.

B. Corporate Commitment and Invested Champions

A distinct advantage our team had on this project was project support from the corporate level down to the system users. Although users had seen some project level efforts to document requirements, they seemed to appreciate the need to document the enterprise requirements in order to associate them with the enterprise systems. Our team found that the cooperative environment made it easy to access project personnel and that they were very forthcoming with supporting information. Often times in anticipation of our meeting personnel would prepare fact sheets, diagrams, or outlines of their organizational processes to facilitate our discussions. At the beginning of this project, we had identified that user cooperation was a potential risk factor. Our efforts to brief personnel on the importance of the project and its potential benefits seemed to mitigate the potential risk in this area.

C. Organizing Requirements

A surprising lesson for our team was the necessity to develop a methodology to organize and bundle the requirements in a logical fashion. Because this requirements documentation effort spanned an entire enterprise, the context of a requirement in isolation could easily be lost. Using the concept of nested requirements, segregation by input, process, analysis, mechanisms, and output as well as annotating related requirements allowed our organization to document requirements and produce meaningful reports. Although much is written about the nature of good requirements, there is little about how to group and relate requirements in a logical manner. Incorporating the IDEF0 approach enabled our team to tackle this challenge.

IX. Conclusion Center

Effective information engineering is the optimal utilization of information systems based upon the originating requirements. The READ project has enabled NAVOCEANO to make informed engineering decisions based upon originating requirements.